IN THE CLAIMS

Please amend the claims as follows. All pending claims after this amendment are listed below for the convenience of the Examiner. Claims amended by the Amendment are indicated as such.

1. In a computer system, a method of detecting input device support of a screen element of a graphical user interface comprising:

examining a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events;

automatically identifying said screen element as supporting said input device when input device-handling program code is associated with said screen element; and

automatically examining a superclass class definition of said screen element's class definition if said class definition of said screen element is not identified as supporting said input device.

2. The method in accordance with claim 1 including the step of marking said screen element when said input device-handling capability is identified.

3. The method in accordance with claim 1 including the step of modifying the look of said screen element when said input device-handling capability is identified.

4. The method in accordance with claim 1 wherein said examining step is performed during a construction process of said screen element.

5. The method in accordance with claim 1 wherein said runtime version of said screen element comprises a version of a class definition of said screen element.

6. The method in accordance with claim 5 including the step of examining a superclass class definition of said screen element's class definition if said class definition of said screen element is not identified as supporting said input device.

7. The method in accordance with claim 1 wherein said examining step comprises examining one or more interface declarations associated with said screen element.

8. The method in accordance with claim 7 wherein said interface declaration is contained in an implements clause.

9. The method in accordance with claim 1 wherein said examining step comprises determining whether said screen element has delegated processing of said input device's input to other program code and identifying said screen element as supporting an input device when said input is so delegated.

10. In a computer system, a method of determining input device support of a screen element of a graphical user interface comprising:

examining a class definition of a screen element of a graphical user interface to detect an ability to process an input device's events; and

automatically identifying said screen element as supporting input device input if said class definition includes a method supporting said input device's input.

11. (Amended) In a computer system, a method of determining input device support of a screen element of a graphical user interface comprising:

examining a class definition of a screen element of a graphical user interface to detect an ability to process an input device's events;

automatically identifying said screen element as supporting input device input if said class definition includes a method supporting said input device's input; and

examining a class definition of a superclass of the screen element if said class definition of said screen element does not include said method.

12. The method in accordance with claim 10 wherein said examining step is performed at runtime.

13. The method in accordance with claim 10 including the step of marking said screen element if said class definition includes a method supporting said input device's input.

14. The method in accordance with claim 10 wherein said examining step includes determining if said screen element has delegated processing of said input device's input to other program code and identifying said screen element as supporting said input device if said processing is so delegated.

15. (Amended) A computer program product comprising a computer usable medium having computer readable program code embodied therein for detecting input device support of a screen element of a graphical user interface comprising:

computer readable program code configured to cause a computer to examine a runtime version of said screen element of a graphical user interface; and computer readable program code configured to cause a computer to automatically identify said screen element as supporting an input device when input device-handling program code is associated with said screen element.

16. The computer program product in accordance with claim 15 wherein said computer readable program code is further configured to mark said screen element when said input device-handling capability is identified.

17. The computer readable program product in accordance with claim 15 wherein said computer readable program code is further configured to examine one or more interface declarations associated with said screen element.

18. A computer comprising:

a display for displaying at least one screen element of a graphical user interface;

at least one input device; and

a detector configured to examine a runtime version of said screen element to automatically identify whether said screen element supports said input device by determining whether input device-handling program code is associated with said screen element.

19. The computer in accordance with claim 18 wherein said detector comprises program code readable by a processor of said computer.

20. The computer in accordance with claim 18 wherein said detector is configured to examining one or more interface declarations associated with said screen element to determine if said code is associated with said screen element.

21. In a computer system, a method of detecting input device support of a screen element of a graphical user interface comprising:

automatically examining a runtime version of code associated with a screen element to detect an ability to process events associated with input devices;

automatically determining at runtime whether said screen element delegated processing of said events associated with said input devices to other program code; and

automatically examining said runtime version of code associated with said screen element to detect a declaration of program code that is indicative of support associated with said screen elements for a given input device.

22. In a computer system, a method of detecting input device support of a screen element of a graphical user interface comprising:

before runtime, examining a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events;

automatically identifying said screen element as supporting said input device when input device-handling program code is associated with said screen element; and

automatically examining a superclass class definition of said screen element's class definition if said class definition of said screen element is not identified as supporting said input device.

REMARKS

The Examiner is thanked for his review of this application. Claims 11 and 15 have been amended. The amended claims 11 and 15 do not introduce new matter or present new issues requiring further consideration or search. Dependent claim 11 was only rewritten to include all of the limitations of independent claim 10. Amendment of claim 11 as proposed places claim 11 in condition for allowance as indicated in the Final Office Action dated April 24, 2002. The amendment of claim 15 is simply to correct a formality by deleting a superfluous comma. Claims 1-22 remain pending after entry of the present amendment.

Rejections under 35 U.S.C. § 103

Claims 10-21 were rejected under 35 U.S.C. 103(a) as being unpatentable over Crutcher et al. ("Crutcher") (U.S. Patent No. 5,844,560) in view of Carey et al. ("Carey") (U.S. Patent No. 6,122,627). These rejections are respectfully traversed.

To establish a *prima facie* case of obviousness, there must be some suggestion or motivation, either in the references or in the knowledge generally available to one having ordinary skill in the art, to combine the references. Additionally, the references when combined must teach or suggest all the claim limitations. As discussed below, the Office has not established a *prima facie* case of obviousness because there is neither suggestion nor motivation, in either the references or in the knowledge of one having ordinary skill in the art at the time of the invention, to have combined the references in the manner proposed. Furthermore, the references when combined do not teach or suggest all of the claim limitations.

Crutcher teaches a graphical user interface (GUI) element (e.g., control button) that has a three-dimensional appearance. The GUI element is capable of performing multiple operations each of which is initiated by activation of a particular region of the GUI element. Activation of a region of the GUI element is performed by controlling a positioning device (e.g., mouse) to locate and activate a positioning element (e.g., mouse cursor) over the region of the GUI element. Activation of the positioning element when located over the GUI element causes the GUI element to generate an appropriate signal. Crutcher teaches a means for monitoring the active status and location of the positioning element to determine whether a region of the GUI element is being activated by the positioning device. As a reaction to activation by the positioning device via the positioning element, the appearance of the GUI element is altered to provide visual feedback that the positioning device action has been performed.

With respect to claim 10, Crutcher does not teach a method of determining input device support of a screen element of a GUI. Rather, Crutcher teaches a response of a GUI element to an input after the input is received from an input device. Furthermore with respect to claim 10, Crutcher does not teach examining a class definition of a screen element of a GUI to detect an ability to process an input device's events. Crutcher teaches a specific method by which a GUI element processes an input device's event after the input event has occurred.

Consider the case in which a user is attempting to activate the GUI element using a mouse as the input device, but the GUI element by design does not support a mouse input device. Without the benefit of the claimed invention, the GUI element will appear normally to the user but will not react to activation by the mouse cursor. Since, the teachings of Crutcher only describe the response of the GUI element after activation by the mouse cursor the user may think that either the program (i.e., GUI element) or the mouse is malfunctioning. However, the reality is that both the program and the mouse are functioning properly. With the claimed invention, the lack of mouse input device support by the GUI element is automatically determined by the computer system and conveyed to the user through visual feedback. Therefore, the user understands prior to attempting activation of the GUI element by the mouse cursor that the GUI element does not support the mouse input device.

With respect to claims 15, 18, and 21, the Examiner asserts the following: "Note that the claim (i.e., claim 1 of Crutcher) is broad and recites that the runtime version of the element is examined and subsequently identified as supporting the input device. This is status indication of the input device, and the element is marked or modified accordingly." The Applicants respectfully disagree with this assertion. The claim does not recite that the runtime version of a

GUI element is examined. Rather, the claim of Crutcher recites "... a means for monitoring the active status and location of the positioning element to determine whether a region of the control button is being pushed by the positioning device ..." Per the embodiments of Crutcher, the positioning element refers to something analogous to a mouse cursor, not a GUI element. The control button is analogous to a GUI element. The claim recites monitoring of the positioning element not monitoring of the control button. Therefore, the claim of Crutcher does not recite that a runtime version of the element (i.e., GUI element) is examined. Furthermore, the assertion that the runtime GUI element is subsequently identified as supporting the input device may be correct for Crutcher, but teaches away from the claimed invention.

The phrase "subsequently identified" refers to the fact that the user must attempt to activate the GUI element with the positioning element via the input device before any knowledge of input device support is obtained. The claimed invention examines a runtime version of the GUI element to automatically identify the capability of the GUI element to support an input device. The claimed invention does not rely on GUI element activation to facilitate observation of the GUI element response in an attempt to identify supported input devices. Rather, the claimed invention considers the input device-handling program code associated with the GUI element to determine if an input device is supported by the GUI element. The user-implemented trial-and-error method for determining if a GUI element is capable of responding to various input devices as inferred by the Examiner with reference to Crutcher is not in any way similar to the method of the claimed invention. Furthermore, contrary to the claimed invention, such a user-implemented trial-and-error method is a reactive method that is not implementable in a computer program.

Carey teaches development of a tool to be used to obtain different specific "views" of data in a relational database management system. Upon receipt of a query referencing a view type, a query engine generates a query plan that builds mock, or proxy, application type objects in computer memory that do not contain data. An application can run methods on the application objects. The proxy objects can redo the same calculations that a view would do to obtain data. In this way, Carey builds objects from view definitions and queries.

The Examiner asserts that Carey shows details of class definitions being associated with GUI elements. Nothing in Carey as referenced by the Examiner has anything to do with details of class definitions being associated with GUI elements. Furthermore, nothing in Carey as referenced by the Examiner has anything to do with examining a runtime version of a screen element of a graphical user interface to detect an ability to process an input device's events and automatically identify the screen element as supporting the input device when input device-handling program code is associated with the screen element. Simply because a GUI element is developed in an object oriented programming language that utilizes classes does not make it obvious to one of ordinary skill in the art to examine the classes to identify methods (i.e., code) that are used to support input devices as claimed by the present invention.

Neither the teachings nor the nature of the problem solved in either Crutcher or Carey, or the combination thereof, motivate or suggest to one of ordinary skill in the art at the time of the invention to combine the reference teachings in a manner that would make the claimed invention obvious. Remembering that the references of Crutcher and Carey must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention, neither Crutcher nor Carey, nor the combination thereof, teach all of the claimed features of claims 10-21. Thus, the Examiner has not established a *prima facie* case of obviousness. For at least these reasons, the Applicants respectfully request that the rejections of independent claims 10, 15, 18, and 21 be withdrawn. For at least the same reasons, the Applicants respectfully submit that dependent claims 12-14, 16-17, and 19-20 are patentable over the cited art of record.

Accordingly, a notice of allowance is respectfully requested. Alternatively, the Applicants submit that the claims, in view of the art of record, are in condition for Appeal. If the

Examiner has any questions concerning the present amendment, the Examiner is kindly requested to contact the undersigned at (408) 749-6900 x6903. If any other fees are due in connection with filing this amendment, the Commissioner is also authorized to charge Deposit Account No. 50-0805 (Order No. SUNMP068). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
MARTINE & PENILLA, LLP

[illegible]

Albert S. Penilla, Esq.
Reg. No. 39,487

MARTINE & PENILLA, LLP
710 Lakeway Drive, Suite 170
Sunnyvale, CA 94085
Telephone: (408) 749-6900
Customer Number 32291